***INSTALLATION***

***django messages***

**Adding a message**[**¶**](https://docs.djangoproject.com/en/4.0/ref/contrib/messages/#adding-a-message)

To add a message, call:

**from** **django.contrib** **import** messages

messages.add\_message(request, messages.INFO, 'Hello world.')

Some shortcut methods provide a standard way to add messages with commonly used tags (which are usually represented as HTML classes for the message):

messages.debug(request, '**%s** SQL statements were executed.' % count)

messages.info(request, 'Three credits remain in your account.')

messages.success(request, 'Profile details updated.')

messages.warning(request, 'Your account expires in three days.')

messages.error(request, 'Document deleted.')

**Displaying messages**[**¶**](https://docs.djangoproject.com/en/4.0/ref/contrib/messages/#displaying-messages)

**get\_messages(*request*)**[**[source]**](https://docs.djangoproject.com/en/4.0/_modules/django/contrib/messages/api/#get_messages)[**¶**](https://docs.djangoproject.com/en/4.0/ref/contrib/messages/#django.contrib.messages.get_messages)

**In your template**, use something like:

{% **if** messages %}

<ul class="messages">

{% **for** message **in** messages %}

<li{% **if** message.tags %} class="{{ message.tags }}"{% endif %}>{{ message }}</li>

{% endfor %}

</ul>

{% endif %}

If you’re using the context processor, your template should be rendered with a **RequestContext**. Otherwise, ensure **messages** is available to the template context.

Even if you know there is only one message, you should still iterate over the **messages** sequence, because otherwise the message storage will not be cleared for the next request.

The context processor also provides a **DEFAULT\_MESSAGE\_LEVELS** variable which is a mapping of the message level names to their numeric value:

{% **if** messages %}

<ul class="messages">

{% **for** message **in** messages %}

<li{% **if** message.tags %} class="{{ message.tags }}"{% endif %}>

{% **if** message.level == DEFAULT\_MESSAGE\_LEVELS.ERROR %}Important: {% endif %}

{{ message }}

</li>

{% endfor %}

</ul>

{% endif %}

**Outside of templates**, you can use **[get\_messages()](https://docs.djangoproject.com/en/4.0/ref/contrib/messages/" \l "django.contrib.messages.get_messages" \o "django.contrib.messages.get_messages)**:

**from** **django.contrib.messages** **import** get\_messages

storage = get\_messages(request)

**for** message **in** storage:

do\_something\_with\_the\_message(message)

2nd template of 3rd entry

from django.http import HttpResponse

from django.shortcuts import render

from sympy import hn1

from . import views

from datetime import datetime

from lunar.models import contact

from django.contrib import messages

def home(request):

    return HttpResponse("this is check for response")

# def home(request):

#     return render(request, 'home.html')

def index(request):

    return render(request, 'index.html')

# def contact(request):

#     return render(request, 'contact.html')

def about(request):

    return render(request, 'about.html')

# def subcontact(request):

#     if  request.method == "post":

#         name = request.post.get('name')

#         email = request.post.get('email')

#         password = request.post.get('password')

#     return render(request, 'subcontact.html')

def contact(request):

    if  request.method == "post":

        name = request.post.get('name')

        email = request.post.get('email')

        phone = request.post.get('phone')

        desc = request.post.get('desc')

        contact=contact(name=name,email=email,phone=phone,desc=desc, date=(datetime.today))

        contact.save()

        messages.success(request, 'your message has been sent')

        return render(request, 'contact.html')

***DJANGO-ADMIN***

Change name of admin page paste it in project urls

admin.site.site\_header = "Custom bookstore admin"

admin.site.site\_title = "Custom bookstore admin site"

admin.site.index\_title = "Custom Bookstore Admin"

***STARTPROJECT***

***STARTAPP***

***URL’S***

from django.contrib import admin

from django.urls import path , include

from . import views

admin.site.site\_header = "TAXALIAN ADVENTURER Admin"

admin.site.site\_title = "TAXALIAN ADVENTURER Admin Portal"

admin.site.index\_title = "Welcome to TAXALIAN ADVENTURER Researcher Portal"

urlpatterns = [

    path('admin/', admin.site.urls),

    path('',views.home,name='home'),

    path('about/',views.about,name='about'),

    path('contact/',views.contact,name='contact'),

    path('index/',views.index,name='index'),

    path('services/',views.services,name='services'),

    path('packages/',views.packages,name='packages'),

    path('partner/',views.partner,name='partner'),

    path('participent/',views.participent,name='participent'),

    path('adventurer/',views.adventurer,name='adventurer'),

    path('prize\_winner/',views.prize\_winner,name='prize\_winner'),

    path('invester/',views.invester,name='invester'),

    path('desired\_area/',views.desired\_area,name='desired\_area'),

    path('training\_learning/',views.training\_learning,name='training\_learning'),

    path('tasks/',views.tasks,name='tasks'),

    path('subcontact/',views.subcontact,name='subcontact'),

    path('earth/',include('earth.urls')),

    path('galaxy/',include('galaxy.urls')),

    path('jupiter/',include('jupiter.urls')),

    path('mars/',include('mars.urls')),

    path('sun/',include('sun.urls')),

    ]

"""computer URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

    https://docs.djangoproject.com/en/3.1/topics/http/urls/

Examples:

Function views

    1. Add an import:  from my\_app import views

    2. Add a URL to urlpatterns:  path('', views.home, name='home')

Class-based views

    1. Add an import:  from other\_app.views import Home

    2. Add a URL to urlpatterns:  path('', Home.as\_view(), name='home')

Including another URLconf

    1. Import the include() function: from django.urls import include, path

    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))

"""

from django.contrib import admin

from django.urls import path , include

from . import views

admin.site.site\_header  =  "COMPUTER\_CPU Admin"

admin.site.site\_title  =  " COMPUTER\_CPU Site"

admin.site.index\_title  =  "COMPUTER\_CPU Admin"

urlpatterns = [

    path('admin/', admin.site.urls),

    path('',views.home, name='home'),

    path('contact/',views.contact,name='contact'),

    path('index/',views.index, name='index'),

    path('intro/',views.intro,name='intro'),

    path('main/',views.main, name='main'),

    path('features/',views.features,name='features'),

    path('MDuniform/',views.MDuniform,name='MDuniform'),

    path('detail/',views.detail, name='detail'),

    path('CPU/', include('CPU.urls')),

]

***APPSURL***

from django.contrib import admin

from django.urls import path,include

from . import views

urlpatterns = [

    path('',views.home,name='home'),

    ]

***VIEWS***

from datetime import date

import datetime

import email

from unicodedata import name

from urllib import request

from django.contrib import messages

from django.http import HttpResponse

from django.shortcuts import render

from . import views

# Create your views here.

def home(request):

    return render(request, 'home.html')

def index(request):

    return render(request, 'index.html')

def contact(request):

    return render(request, 'contact.html')

def about(request):

    return render(request, 'about.html')

def services(request):

    return render(request, 'services.html')

def partner(request):

    return render(request, 'partner.html')

def participent(request):

    return render(request, 'participent.html')

def packages(request):

    return render(request, 'packages.html')

def tasks(request):

    return render(request, 'tasks.html')

def adventurer(request):

    return render(request, 'adventurer.html')

# def base(request):

#     return render(request,'base.html')

def desired\_area(request):

    return render(request, 'desired\_area.html')

def discounts(request):

    return render(request, 'discounts.html')

def expertises(request):

    return render(request, 'expertises.html')

def invester(request):

    return render(request, 'invester.html')

def prize\_winner(request):

    return render(request, 'prize\_winner.html')

def procedures(request):

    return render(request, 'procedures.html')

def techniques(request):

    return render(request, 'techniques.html')

def training\_learning(request):

    return render(request, 'training\_learning.html')

def subcontact(request):

    if  request.method == "post":

        name = request.post.get('name')

        email = request.post.get('email')

        password = request.post.get('password')

    return render(request, 'subcontact.html')

# def post(request):

# if  request.method == "post":

#     name = request.post.get('name')

#     email = request.post.get('email')

#     phone = request.post.get('phone')

#     desc = request.post.get('desc')

#     contact=contact(name=name,email=email,phone=phone,desc=desc, date = (datetime.today))

#     contact.save()

#     messages.success(request, 'your message has been sent')

#     return render(request, 'contact.html')

2nd template of 3rd entry

from django.http import HttpResponse

from django.shortcuts import render

from sympy import hn1

from . import views

from datetime import datetime

from lunar.models import contact

from django.contrib import messages

def home(request):

    return HttpResponse("this is check for response")

# def home(request):

#     return render(request, 'home.html')

def index(request):

    return render(request, 'index.html')

# def contact(request):

#     return render(request, 'contact.html')

def about(request):

    return render(request, 'about.html')

# def subcontact(request):

#     if  request.method == "post":

#         name = request.post.get('name')

#         email = request.post.get('email')

#         password = request.post.get('password')

#     return render(request, 'subcontact.html')

def contact(request):

    if  request.method == "post":

        name = request.post.get('name')

        email = request.post.get('email')

        phone = request.post.get('phone')

        desc = request.post.get('desc')

        contact=contact(name=name,email=email,phone=phone,desc=desc, date=(datetime.today))

        contact.save()

        messages.success(request, 'your message has been sent')

        return render(request, 'contact.html')

from re import A

from ssl import Purpose

from django.http import HttpResponse

from django.shortcuts import render

from . import views

from CPU.models import uniform

def home(request):

    return render(request,'home.html')

def index(request):

    return render(request,'index.html')

def MDuniform(request):

    n = ''

    if request.method=='POST':

        shirt = request.POST.get('shirt')

        Banyan  = request.POST.get( 'Banyan')

        pent     = request.POST.get( 'pent')

        underware  = request.POST.get('underware')

        sweater    = request.POST.get('sweater')

        tie        = request.POST.get('tie')

        data = uniform(shirt=shirt,Banyan=Banyan,pent=pent,underware=underware,sweater=sweater,tie=tie)

        data.save()

        n = 'aap ka data ja chuka hey wahan check karain bhai samjhay data inserted'

    return render(request,"contact.html",{'n':n})

def contact(request):

    return render(request,'contact.html')

# def text(request):

#     if request.method == 'GET':

#        text=request.GET.get('text')

#        data={'text': text}

#     return render(request,'text.html',data)

# def text(request):

#     Atext=request.GET.get('text','default')

#     rpu=request.GET.get('rpu','off')

#     print(rpu)

#     print(Atext)

#     return HttpResponse("rpu")

def text(request):

    Atext=request.GET.get('text','default')

    rpu=request.GET.get('rpu','off')

    upper=request.GET.get('upper','off')

    # print(rpu)

    # print(Atext)

    if rpu == "on" :

        # analyzed=Atext

        punctuations = ''' ? ` , : , @  ; ,% ^ <<,,,>>>..!!!| “ ”, \_ - ( ), { },[ ] ,  ,\*, \\\::"""'  ~ , / , , #,$, &  '''

        analyzed=""

        for i in Atext:

            if i not in punctuations:

               analyzed=analyzed+i

        data={'rpu': rpu , 'btext': analyzed }

        return render(request,'text.html', data)

    elif(upper == "on") :

        punctuations = '''?`,:,@;,%^<<,,,>>>{..!!!|“”,\_-(),,[],,\*,\\\}::"""'~,/,,#,$,&'''

        analyzed=""

        for i in Atext:

            if i not in punctuations:

               analyzed=analyzed + i.upper()

        data={'rpu': rpu , 'btext': analyzed }

        return render(request,'text.html', data)

    else:

        return HttpResponse(" sab ghalat kar diya khaotay tou hay hi AQAL say paidal ")

def intro(request):

    return render(request,'intro.html')

def main(request):

    return render(request,'main.html')

def features(request):

    return render(request,'features.html')

def detail(request):

    return render(request,'detail.html')

***APPSVIEWS***

from django.http import HttpResponse

from django.shortcuts import render

from . import views

# Create your views here.

def home(request):

    data = {'v' : "THIS IS MY CITY VIEW"}

    return render(request,'earth/home.html', data )

***HTML’S TOP***

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>رابطہ کریں</title>

</head>

<body>

    {% extends 'base.html' %}

    {% block title %} Training and Learning {% endblock title %}

    {% block body %}

    <!--

    <body >

        <div class="container">

        </div>

        <div class="container-fluid my-1 px=0 mb-5 "></div>

        <h1 class="text-center"> THIS IS OUR CONTACT PAGE </h1>

        <h1 class="text-center"> ﴾۩ ﷽ ۩﴿

        </h1>

        <h1 class="text-center"> WELL COME

        </h1>

        <h2 class="text-center"> یہ ہمارے اور آپ کے تعاون اور محبت کا ذریعہ ہے </h2>

        <h2 class="text-center"> !!!پیار بڑھانے کے لیے اس نمبر پر رابطہ کریں !!!03453565169 !!! </h2>

    </body>

 -->

    <div class="container-fluid px=0 mb-5 ">

        <!-- <h5 class="text-center"> Contact No 03453565169 </h5> -->

        <div id="carouselExampleControls" class="carousel slide" data-bs-ride="carousel">

            <div class="carousel-inner">

                <div class="carousel-item active">

                    <img src="/static/55.JPG" class="d-block w-100 " alt="..." />

                </div>

                <div id="carouselExampleControls" class="carousel slide" data-bs-ride="carousel">

                    <div class="carousel-inner">

                        <div class="carousel-item active">

                            <img src="/static/57.JPG" class="d-block w-100 " alt="..." />

                        </div>

                        <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleControls"

                            data-bs-slide="prev">

                            <span class="carousel-control-prev-icon" aria-hidden="true"></span>

                            <span class="visually-hidden">Previous</span>

                        </button>

                        <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleControls"

                            data-bs-slide="next">

                            <span class="carousel-control-next-icon" aria-hidden="true"></span>

                            <span class="visually-hidden">Next</span>

                        </button>

                    </div>

                    <div class="carousel-item">

                        {% comment %} <img src="/static/14.JPG " class="d-block w-100" alt="..."> {% endcomment %}

                        <!-- <img src="/static/51.JPG " class="d-block w-100" alt="..."> -->

                        <img src="/static/56.JPG" class="d-block w-100 " alt="..." />

                        <div class="carousel-item" align="center">

                        </div>

                    </div>

                    <div class="carousel-item">

                        <!-- <img src="/static/14.JPG" class="d-block w-100" alt="..." /> -->

                        <!-- <img src="/static/50.JPG" class="d-block w-100" alt="..." /> -->

                    </div>

                    <!-- <div class="mb-3">

                <label for="exampleFormControlInput1" class="form-label">Email address</label>

                <input type="email" class="form-control" id="exampleFormControlInput1" placeholder="name@example.com">

              </div>

              <div class="mb-3">

                <label for="exampleFormControlTextarea1" class="form-label">Example textarea</label>

                <textarea class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>

              </div> -->

              <div class="container-fluid px=0 mb-3 ">

                {% csrf\_token %}

                </div>

                <div class="container ">

                    <h1 class="text-center">ہم سے رابطہ کیجیۓ اور شرم بالکل محسوس نہ کیجیۓ</h1>

                    <form method="post" action="/subcontact">

                        <div class="form-group">

                            <label for="name" >Name</label>

                            <input type="text" class="form-control" id="name" name="name" placeholder="enter your name">

                            <div id="emailHelp" class="form-text"> </div>

                        </div>

                        <div class="mb-3">

                            <label for="exampleInputEmail1" class="form-label">Email address</label>

                            <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">

                            <div id="emailHelp" class="form-text">ڈرنا نہیں ہے ہم کسی کو نہیں بتاتے</div>

                          </div>

                        <div class="mb-3">

                            <label for="exampleInputPassword1" class="form-label">Password</label>

                            <input type="password" class="form-control" id="exampleInputPassword1">

                        </div>

                        <div class="mb-3 form-check">

                            <input type="checkbox" class="form-check-input" id="exampleCheck1">

                            <label class="form-check-label" for="exampleCheck1">Check me out</label>

                        </div>

                        <button type="submit" class="btn btn-primary">Submit</button>

                    </form>

                </div>

                {% endblock body %}

</body>

</html>

***MANAGE.PY***

import os

import sys

def main():

    """Run administrative tasks."""

    os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'universe.settings')

    try:

        from django.core.management import execute\_from\_command\_line

    except ImportError as exc:

        raise ImportError(

            "Couldn't import Django. Are you sure it's installed and "

            "available on your PYTHONPATH environment variable? Did you "

            "forget to activate a virtual environment?"

        ) from exc

    execute\_from\_command\_line(sys.argv)

if \_\_name\_\_ == '\_\_main\_\_':

    main()

2ND MANAGE.PY

#!/usr/bin/env python

"""Django's command-line utility for administrative tasks."""

import os

import sys

def main():

    """Run administrative tasks."""

    os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'computer.settings')

    try:

        from django.core.management import execute\_from\_command\_line

    except ImportError as exc:

        raise ImportError(

            "Couldn't import Django. Are you sure it's installed and "

            "available on your PYTHONPATH environment variable? Did you "

            "forget to activate a virtual environment?"

        ) from exc

    execute\_from\_command\_line(sys.argv)

if \_\_name\_\_ == '\_\_main\_\_':

    main()

***APPSHTML***

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>earth\_home</title>

    {% load static%}

    <link rel="stylesheet" href="{% static 'st.css'%}">

</head>

<body bgcolor="cyan">

    <h1 class="text-center">Assalamualaikum</h1>

    <h2 class="  text-center"> THIS IS OUR HOME PAGE OF EARTH APP </h2>

    <H1 class="text-center">WELL COME</H1>

<h1 class="st" > !!!!!  ہمارے گھر میں آپ سب کو بہت بہت خوش آمدید!!!!! </h1>

                {{v}}

</body>

</html>

***MODELS.PY MODELS ADMIN.PY***

from distutils.command.upload import upload

import email

from email.policy import default

from unicodedata import name

from django.db import models

from matplotlib import image

from django.contrib import admin

# Register your models here.

from .models import car

admin . site . register(car)

from .models import health

admin . site . register(health)

# Create your models here.

class Membership(models.Model):

    person = models.CharField(max\_length=50)

    group = models.CharField(max\_length=50)

    invite\_reason = models.CharField(max\_length=64)

    images = models.ImageField(upload\_to = "images" ,default="")

    def \_\_str\_\_(self):

        return self.person

class info(models.Model):

    name = models.CharField(max\_length=50)

    email = models.CharField(max\_length=50)

    password = models.CharField(max\_length=30)

    def \_\_str\_\_(self):

        return self.name

class contact(models.Model):

    name = models.CharField(max\_length=50)

    email = models.CharField(max\_length=50)

    phone = models.CharField(max\_length=50)

    desc = models.TextField()

    date = models.DateField()

    def \_\_str\_\_(self):

        return self.name

from django.db import models

class car(models.Model):

    engine =models.CharField(max\_length=50,default="")

    tyres =models.CharField(max\_length=50,default="")

    doors =models.CharField(max\_length=50,default="")

    mirrors =models.CharField(max\_length=50,default="")

    def \_\_str\_\_(self):

        return self.engine

class health(models.Model):

    body  =models.CharField(max\_length=50,default="")

    chest =models.CharField(max\_length=50,default="")

    muscles =models.CharField(max\_length=50,default="")

    def \_\_str\_\_(self):

        return self.body

***APPS.PY***

from django.apps import AppConfig

class EarthConfig(AppConfig):

    default\_auto\_field = 'django.db.models.BigAutoField'

    name = 'earth'

***TESTS.PY***

from django.test import TestCase

# Create your tests here.

***ADMIN.PY***

from distutils.log import info

from django.contrib import admin

from .models import Membership

from .models import info

admin . site . register (Membership)

admin . site . register (info)

# Register your models here.

***STATIC ST.CSS***

.st{

   color: blue;

}

{% extends 'base.html' %}

{% block title %}{% endblock title %}

{% block body %} {% endblock body %}

{% comment %}  {% endcomment %}

#added manually

STATICFILES\_DIRS = [os.path.join(BASE\_DIR,"static"),

]

# [os.path.join(BASE\_DIR,"templates")],

***ADMINPAGE***

***SETTINGS.PY***

from email import message

import os

from pathlib import Path

from django.contrib.messages import constants as messages

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

from email import message

import os

from pathlib import Path

from django.contrib.messages import constants as messages

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'earth.apps.EarthConfig',

    'galaxy.apps.GalaxyConfig',

    'sun.apps.SunConfig',

    'jupiter.apps.JupiterConfig',

    'mars.apps.MarsConfig',

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [os.path.join(BASE\_DIR,"templates")],

        'APP\_DIRS': True ,

        'OPTIONS': {

            'context\_processors': [

                'django.template.context\_processors.debug',

                'django.template.context\_processors.request',

                'django.contrib.auth.context\_processors.auth',

                'django.contrib.messages.context\_processors.messages',

            ],

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': os.path.join(BASE\_DIR , 'db.sqlite3'),

    }

}

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.1/howto/static-files/

STATIC\_URL = '/static/'

# added manually

STATICFILES\_DIRS = [os.path.join(BASE\_DIR,"static")]

////////////////////2nd ////////////////////database//////////////////////

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.postgresql',

        'NAME': 'DB1',

        'USER': 'postgres',

        'PASSWORD': 'rana',

        'HOST': 'localhost',

    }

}

"""

Django settings for computer project.

Generated by 'django-admin startproject' using Django 3.1.2.

For more information on this file, see

https://docs.djangoproject.com/en/3.1/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/3.1/ref/settings/

"""

import os

from pathlib import Path

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'q=sha5p#zd)^lxs%b^97k3nq!2#w1a758kr)i(f(knjwj2qg30'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'CPU.apps.CpuConfig',

]

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'computer.urls'

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [os.path.join(BASE\_DIR,'templates')],

        'APP\_DIRS': True,

        'OPTIONS': {

            'context\_processors': [

                'django.template.context\_processors.debug',

                'django.template.context\_processors.request',

                'django.contrib.auth.context\_processors.auth',

                'django.contrib.messages.context\_processors.messages',

            ],

        },

    },

]

WSGI\_APPLICATION = 'computer.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': BASE\_DIR / 'db.sqlite3',

    }

}

# Password validation

# https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

    {

        'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

    },

]

# Internationalization

# https://docs.djangoproject.com/en/3.1/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.1/howto/static-files/

STATIC\_URL = '/static/'

# added manually

STATICFILES\_DIRS = os.path.join(BASE\_DIR,"static"),

***ASGI.PY***"""

ASGI config for computer project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see

https://docs.djangoproject.com/en/3.1/howto/deployment/asgi/

"""

import os

from django.core.asgi import get\_asgi\_application

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'computer.settings')

application = get\_asgi\_application()

***WSGI.PY***

"""

WSGI config for computer project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see

https://docs.djangoproject.com/en/3.1/howto/deployment/wsgi/

"""

import os

from django.core.wsgi import get\_wsgi\_application

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'computer.settings')

application = get\_wsgi\_application()

***NAVBAR***

***/PICS***

 <div class="container">

      </div>

      <div class="carousel-item">

        <img src="/static/14.JPG" class="d-block w-100" alt="..." />

        </div>

      <div class="carousel-item">

        {% comment %} <img src="/static/14.JPG " class="d-block w-100" alt="..."> {% endcomment %}

        <img src="/static/18.JPG " class="d-block w-100" alt="...">

        <div class="carousel-item" align="center">

        </div>

      </div>

      <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleControls"

        data-bs-slide="prev">

        <span class="carousel-control-prev-icon" aria-hidden="true"></span>

        <span class="visually-hidden">Previous</span>

      </button>

      <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleControls"

        data-bs-slide="next">

        <span class="carousel-control-next-icon" aria-hidden="true"></span>

        <span class="visually-hidden">Next</span>

      </button>

    </div>

NAVBARS/ CAROUSALS/

PICS

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

</head>

<title>base</title>

<h1 class="text-center">!ASSALAMUALAIKUM!</h1>

<head>

  <!-- Required meta tags -->

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.rtl.min.css"

    integrity="sha384-+qdLaIRZfNu4cVPK/PxJJEy0B0f3Ugv8i482AKY7gwXwhaCroABd086ybrVKTa0q" crossorigin="anonymous">

</head>

<body>

  <body>

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">

      <div class="container-fluid my-0.5 ">

        <a class="navbar-brand" href="/contact">Navbar</a>

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"

          aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

          <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">

          <ul class="navbar-nav me-auto mb-2 mb-lg-0">

            <li class="nav-item">

              <a class="nav-link active" aria-current="page" href="#">Home</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="/contact">Link</a>

            </li>

            <li class="nav-item dropdown">

              <a class="nav-link dropdown-toggle" href="/contact" id="navbarDropdown" role="button"

                data-bs-toggle="dropdown" aria-expanded="false">

                Dropdown

              </a>

              <ul class="dropdown-menu" aria-labelledby="navbarDropdown">

                <li><a class="dropdown-item" href="/contact">Action</a></li>

                <li><a class="dropdown-item" href="/contact">Another action</a></li>

                <li>

                  <hr class="dropdown-divider">

                </li>

                <li><a class="dropdown-item" href="/contact">Something else here</a></li>

              </ul>

            </li>

          </ul>

          <form class="d-flex">

            <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

            <button class="btn btn-outline-success" type="submit">Search</button>

          </form>

        </div>

      </div>

    </nav>

    <div class="container-fluid mb-1 my-1 px-1 ">

      <div class="carousel-inner">

        <div class="carousel-item active">

          <img src="/static/62.JPG  " class="d-block w-100" alt="...">

        </div>

        <div class="carousel-item">

          <img src="/static/61.JPG" class="d-block w-100" alt="...">

        </div>

        <div class="carousel-item">

          <img src="/static/60.JPG" class="d-block w-100" alt="...">

        </div>

      </div>

      <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleIndicators"

        data-bs-slide="prev">

        <span class="carousel-control-prev-icon" aria-hidden="true"></span>

        <span class="visually-hidden">Previous</span>

      </button>

      <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators"

        data-bs-slide="next">

        <span class="carousel-control-next-icon" aria-hidden="true"></span>

        <span class="visually-hidden">Next</span>

      </button>

    </div>

    </div>

    </div>

***SETTINGS.PY***

"""

Django settings for rabees project.

Generated by 'django-admin startproject' using Django 3.1.2.

For more information on this file, see

https://docs.djangoproject.com/en/3.1/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/3.1/ref/settings/

"""

from email import message

import os

from pathlib import Path

from django.contrib.messages import constants as messages

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = '(y(l2tsqcjm\*&5sxqxv3uu!s6l-0)m==d)=2m7zs+7lu0)@0@!'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'milkyway.apps.MilkywayConfig',

]

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'rabees.urls'

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [os.path.join (BASE\_DIR,"templates")] ,

        'APP\_DIRS': True,

        'OPTIONS': {

            'context\_processors': [

                'django.template.context\_processors.debug',

                'django.template.context\_processors.request',

                'django.contrib.auth.context\_processors.auth',

                'django.contrib.messages.context\_processors.messages',

            ],

        },

    },

]

WSGI\_APPLICATION = 'rabees.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': os.path.join(BASE\_DIR , 'db.sqlite3'),

    }

}

Settings/postgresql/dbsqlite3

# DATABASES = {

#     'default': {

#         'ENGINE': 'django.db.backends.postgresql',

#         'NAME': 'DB1',

#         'USER': 'postgres',

#         'PASSWORD': 'rana',

#         'HOST': 'localhost',

#     }

# }

# Password validation

# https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

    {

        'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

    },

]

# Internationalization

# https://docs.djangoproject.com/en/3.1/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.1/howto/static-files/

STATIC\_URL = '/static/'

# added manually

STATICFILES\_DIRS = [os.path.join(BASE\_DIR,"static")]

"""

Django settings for rabees project.

Generated by 'django-admin startproject' using Django 3.1.2.

For more information on this file, see

https://docs.djangoproject.com/en/3.1/topics/settings/

For the full list of settings and their values, see

https://docs.djangoproject.com/en/3.1/ref/settings/

"""

from email import message

import os

from pathlib import Path

from django.contrib.messages import constants as messages

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = '(y(l2tsqcjm\*&5sxqxv3uu!s6l-0)m==d)=2m7zs+7lu0)@0@!'

# SECURITY WARNING: don't run with debug turned on in production!

DEBUG = True

ALLOWED\_HOSTS = []

# Application definition

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'milkyway.apps.MilkywayConfig',

]

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'rabees.urls'

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [os.path.join (BASE\_DIR,"templates")] ,

        'APP\_DIRS': True,

        'OPTIONS': {

            'context\_processors': [

                'django.template.context\_processors.debug',

                'django.template.context\_processors.request',

                'django.contrib.auth.context\_processors.auth',

                'django.contrib.messages.context\_processors.messages',

            ],

        },

    },

]

WSGI\_APPLICATION = 'rabees.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': os.path.join(BASE\_DIR , 'db.sqlite3'),

    }

}

# Password validation

# https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators

AUTH\_PASSWORD\_VALIDATORS = [

    {

        'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

    },

]

# Internationalization

# https://docs.djangoproject.com/en/3.1/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.1/howto/static-files/

STATIC\_URL = '/static/'

# added manually

STATICFILES\_DIRS = [os.path.join(BASE\_DIR,"static")]

Buttons

btn btn-secondary btn-lg

type="button" class="text center"

***SEND DATA TO MODELS buttons***

<textarea name="text" id="text" cols="80" rows="10" style="height: 112px; width: 744px;"></textarea>

<button style="background-color: darkgreen;

               color: rgb(34, 50, 197);

               border: crimson;

               background: rgb(226, 160, 17);

               " > tajarbah</button>

<button style="border: crimson;"  ></button>

<button style="background: darkblue;" ></button>

<br>

<style>   </style>

<button

style="color: coral;"

 ></button>

<br>

<button style="background-color: blueviolet;">analyze</button>

</html>

div class="container mb-1 my-1 px-10">

        <H1 class="text-center" > CONTACT US </H1>

      <form method="POST" action="/index">

    </div>

    <div class="form-group">

        <select class="form-select" id="floatingSelectGrid" aria-label="Floating label select example">

          <option selected>Open this select menu</option>

          <option value="1">enter your size of chest</option>

          <option value="2">enter your size of body</option>

          <option value="3">enter your size of muscles</option>

        </select>

        <label for="floatingSelectGrid">Works with selects</label>

      </div>

    </div>

    </div>

    <div class="form-group">

      <label for="body"> <strong> body</strong> </label>

      <input type="text"  class="form-control" id="body" name="body" placeholder="enter your body size" >

    </div>

    <div class="form-group">

      <label for="muscles"> <strong> muscles</strong> </label>

      <input type="text"  class="form-control" id="muscles" name="muscles" placeholder="enter your muscles size" >

    </div>

    <div class="form-group">

      <label for="chest"> <strong> chest</strong> </label>

      <input type="text"  class="form-control" id="chest" name="chest" placeholder="enter your chest size" >

    </div>

    <button class="btn btn-danger my-2 text-center" type="submitt">SUBMITT</button>

    <br><br>

    </form>

<div class="alert alert-warning alert-dismissible fade show" role="alert">

<strong>Holy guacamole!</strong> You should check in on some of those fields below.

<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

</div>

***FORM/ MESSAGE HAS BEEN SENT***

<div class="container mb-1 my-1 px-10">

        <H1 class="text-center" > <strong> CONTACT US </strong></H1>

      <form method="post" action="/index">

      {% csrf\_token %}

    {% for message in messages  %}

    {% if messages %}

      <div class="alert alert-warning alert-dismissible fade show" role="alert">

        <strong>ASSALAMUALAIKUM!</strong> Your message has been sent!

        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

      </div>

    <strong>  your message has been sent </strong>

    <div class="alert alert- {{message.tags}} alert-dismissible fade show my-0 " role="alert">

              {{message}}

    <svg class="bi flex-shrink-0 me-2" width="24" height="24" role="img" aria-label="Warning:"><use xlink:href="#exclamation-triangle-fill"/></svg>

    <span  aria-hidden="true" >&times; </span>

    {% endif  %}

    {% endfor %}

    </div>

    <div class="form-group">

      <label for="body"> <strong> body</strong> </label>

      <input type="text"  class="form-control" id="body" name="body" placeholder="enter your body size" >

    </div>

    <div class="form-group">

      <label for="muscles"> <strong> muscles</strong> </label>

      <input type="text"  class="form-control" id="muscles" name="muscles" placeholder="enter your muscles size" >

    </div>

    <div class="form-group">

      <label for="chest"> <strong> chest</strong> </label>

      <input type="text"  class="form-control" id="chest" name="chest" placeholder="enter your chest size" >

    </div>

    <button class="btn btn-danger my-2 text-center" type="submitt">SUBMITT</button>

    <br><br>

    </form>

Punctuation marks

**.**  period (US) or full stop (UK)

**,**  comma

**!**  exclamation mark

**?**  question mark

**:**  colon

**;**  semicolon

**“ ”**  quotation marks

**‘ ’**  single quotes

**( )**   parentheses

**{ }**  curved brackets

**[ ]**   square brackets

**@**    “at” sign or symbol

**\***   asterisk

**~**  tilde

**/**   slash

**–**   dash

**#**   pound sign

**$**   dollar sign

**&** and or ampersand

**.**  period (US) or full stop (UK)

**,**  comma

**!**  exclamation mark

**?** , **: ,**  **; ,**  **“ ”, ‘ ’, ( ),** **{ },[ ]** , **@** ,**\*,**   **~ ,** **/** ,  **–** , **#,$,** **&**

   punctuations = ''' ?  , : ,   ; ,% ^ <<,,,>>>..!!!| “ ”, \_ - ( ), { },[ ] , @ ,\*, \\\::"""'  ~ , / , , #,$, &  '''

remove punctuation function

from email.policy import default

from string import punctuation

from django.http import HttpResponse

from django.shortcuts import render

from sympy import stationary\_points

from jupiter.models import result

from . import views

def home(request):

    return render(request,'home.html')

def index(request):

    return render(request,'index.html')

def about(request):

    return render(request,'about.html')

def book(request):

    b=''

    if request.method == 'POST':

        school=request.POST.get('school')

        physics=request.POST.get('physics')

        chemistry=request.POST.get('chemistry')

        math=request.POST.get('math')

        bio=request.POST.get('bio')

        data=result(school=school,physics=physics,chemistry=chemistry,math=math,bio=bio)

        data.save()

        b='your data has been sent sir you can go and check'

    return render(request,"about.html",{'b':b})

# def text(request):

#     if request.method == 'GET':

#        text=request.GET.get('text')

#        data={'text': text}

#     return render(request,'text.html',data)

# def text(request):

#     Atext=request.GET.get('text','default')

#     rpu=request.GET.get('rpu','off')

#     print(rpu)

#     print(Atext)

#     return HttpResponse("rpu")

def text(request):

    Atext=request.GET.get('text','default')

    rpu=request.GET.get('rpu','off')

    # print(rpu)

    # print(Atext)

    if rpu == "on" :

        # analyzed=Atext

        punctuations = ''' ?  , : ,   ; ,% ^ <<,,,>>>..!!!| “ ”, \_ - ( ), { },[ ] , @ ,\*, \\\::"""'  ~ , / , , #,$, &  '''

        analyzed=""

        for char in Atext:

            if char not in punctuations:

               analyzed=analyzed+char

        data={'rpu': rpu , 'btext': analyzed }

        return render(request,'text.html', data)

    else:

        return HttpResponse(" sab ghalat kar diya khaotay tou hay hi AQAL say paidal ")

ABOUT.HTML

<html>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>about page</title>

</head>

<body bgcolor="buldywood">

    <h1>This about page is going to a zabardast grammer page</h1>

  <body>

    <h1>{{b}}</h1>

    <form  method="GET" action="/text">

<textarea name="text" id="text" cols="80" rows="10" style="height: 112px; width: 744px;"></textarea>

<br>

<input type="checkbox" name="rpu"> remove punctuation

<br>

<button type="submit">analyze the text</button>

<!--

<br>

<label class="container">remove punctuation

    <input type="checkbox" >

    <span class="checkmark"></span>

  </label>

  <label class="container">remove new line

    <input type="checkbox">

    <span class="checkmark"></span>

  </label>

  <label class="container">remove symbols

    <input type="checkbox">

    <span class="checkmark"></span>

  </label>

  <label class="container">remove characters

    <input type="checkbox">

    <span class="checkmark"></span>

  </label>

    <br>

      <button type="submit"   style="background-color: blueviolet;">edit text</button>

    <br> -->

</form>

</html>

TEXT.HTML

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>TEXT FANKARI</title>

</head>

<body bgcolor="darkbrown">

    <div class="container">

    <!-- <h1 class="text-center">!!!!!ASSALAMUALAIKUM!!!!!</h1> -->

    <h1 class="text-center" style="

    position: absolute;

    left: 290px;

    top : 0px;

">!!!!!ASSALAMUALAIKUM!!!!!</h1>

<br><br>

    <h1 >you are on this page to see text fankaariyan</h1>

</div>

</body>

</html>

{{text}}

<H1>yeh asal main khali punctuation button hey agar on hey to punctuation naen hain ok !!!!  han kiy hey ??? {{ rpu }}</H1>

 <h1>aor yeh .... yeh bhi text hey jo likha hey  {{ btext }} </h1>

VIEWS.PY

from email.policy import default

from string import punctuation

from django.http import HttpResponse

from django.shortcuts import render

from sympy import stationary\_points

from jupiter.models import result

from . import views

def home(request):

    return render(request,'home.html')

def index(request):

    return render(request,'index.html')

def about(request):

    return render(request,'about.html')

def book(request):

    b=''

    if request.method == 'POST':

        school=request.POST.get('school')

        physics=request.POST.get('physics')

        chemistry=request.POST.get('chemistry')

        math=request.POST.get('math')

        bio=request.POST.get('bio')

        data=result(school=school,physics=physics,chemistry=chemistry,math=math,bio=bio)

        data.save()

        b='your data has been sent sir you can go and check'

    return render(request,"about.html",{'b':b})

# def text(request):

#     if request.method == 'GET':

#        text=request.GET.get('text')

#        data={'text': text}

#     return render(request,'text.html',data)

# def text(request):

#     Atext=request.GET.get('text','default')

#     rpu=request.GET.get('rpu','off')

#     print(rpu)

#     print(Atext)

#     return HttpResponse("rpu")

def text(request):

    Atext=request.GET.get('text','default')

    rpu=request.GET.get('rpu','off')

    # print(rpu)

    # print(Atext)

    if rpu == "on" :

        # analyzed=Atext

        punctuations = ''' ?  , : ,   ; ,% ^ <<,,,>>>..!!!| “ ”, \_ - ( ), { },[ ] , @ ,\*, \\\::"""'  ~ , / , , #,$, &  '''

        analyzed=""

        for char in Atext:

            if char not in punctuations:

               analyzed=analyzed+char

        data={'rpu': rpu , 'btext': analyzed }

        return render(request,'text.html', data)

    else:

        return HttpResponse(" sab ghalat kar diya khaotay tou hay hi AQAL say paidal ")